Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Environmental Sciences 11 (2011) 334 – 338

Procedia Environmental Sciences

Data Management Trust System in Pervasive Environments

Hu Zhao[a], Sangen Wang[b]

College of Engineering and Technology, Southwest University, SWU,Chongqing, China

[a]tiger100118@163.com, [b]wangsg@swu.edu.cn

Abstract

In this paper we discuss related work from literature in security and distributed data management in general, and mobile security and peer-to-peer collaborative processes in specific. Significant work has been done to address security for various aspects of networking, data storage, device security, etc. for wired networks or networks with centralized or federated control. We compare and contrast the security requirements for resource sharing in pervasive ecosystems, and provide brief descriptions of related work with a focus on mobile devices.

Keywords: Data Management; Trust system; Pervasive System; peer-to-peer; data storage

1. Introduction

A myriad of services are available via wireless interfaces throughout the pervasive environment. Use cases for pervasive environments are slowly evolving from client-server to collaborative peering relations. As computing becomes pervasive, people expect to access services and information anytime and anywhere. These systems lack centralized control, are not under any single administrative domain, and in addition their users are not all known a-priori. Moreover, devices are only guaranteed to be able to communicate with peers in their vicinity – Internet connectivity is 'limited.'

Ranganathan[1] has identified five security problems that arise out of inherent properties of pervasive environments, viz., (1) device authentication, (2) privacy, (3) trust management, (4) device assurance and, (5) recourse. Hubaux et al.[2] classify threats for MANETs at the networking layer to be of two types, viz (1) attacks on the cooperative network layer routing substrate, and (2) attacks on the security mechanisms protecting the MANET itself. We address only the trust, reputation and data management, and also address long term accountability (recourse).

Security for pervasive environments, is about what particular threats need to be addressed, depending on the threat model. These threats are different depending on the application scenario, e.g., a battlefield scenario or a civilian airport terminal. Typically we will be dealing with civilian scenarios, though some of the techniques we are proposing will be applicable to military scenarios as well

2. Nature and Composition of Pervasive environments

Elements constituting a resource rich environment are mobile devices like car computers, wearable com-puters, handheld devices, laptops etc. Personal area networks (PANs) or body-area networks composed of wearable devices are an example of pre-configured networked portable devices and can be abstracted as a single entity – a set of devices collaborating to perform the user's tasks – working towards a common goal. Personal devices function to serve the goals and tasks of the individual users, whereas the devices embedded in the infrastructure providing useful services like alerts, interfaces to sensor networks, location information, weather, traffic conditions, etc., constitute the resource-rich environment.

We envision that portable devices and other ambient devices embedded in the physical infrastructure of pervasive environments – of various forms and functions, will largely outnumber individual users. Computational effort and storage capacities will seldom be in short supply. Micro/nano-sensor networks, temporary community storage/staging

doi:10.1016/j.proenv.2011.12.053

areas will be available for use by mobile devices. However finding reliable and relevant information and services in a timely manner, in this data-intensive and resource-rich environment, will be a challenge.

Smart homes and smart offices are commonly professed examples since they typically have network infrastructures. However, even with minimal networking support, pervasive environments can exist in freeways and public places like beaches and parks. In the near future most of the basic day-to-day activities regardless of location will be augmented by autonomously functioning portable devices. Although much of the technology required to build such systems is cheaply available today – portable computing devices, abundant miniature storage, low power wireless communication, and energy conserving processors – several challenges lie in making this vision a viable reality.

A single device or an ensemble of mobile devices (e.g., wearable or portable computers embedded in apparel or accessories), should be able to dynamically identify and authenticate each other with minimal user intervention.

3. Architecture

As shown in figure V.1, there are essentially three components on the client side(mobile device) viz. the Policy Manager, Context Manager, and the Policy Enforcer. The Beacon is located on a local network device possibly co-located with the wireless base station. The server hosts the Policy Server and the Policy Engine. The beacon periodically broadcasts heartbeats that the sentient Context Manager on the device continuously monitors.The role of the Policy enforcer is to enforce the currently selected policy, whereas the Context Manager is responsible for monitoring the context of the device and selecting the appropriate current policy. The device boots up with an initial default policy. The context manager listens for updates from the policy server. The device can be transit between the home network and other known or unknown networks. Beacons are deployed across the wireless networks, which periodically broadcasts heartbeats. The context manager listens for these heartbeats and based on the information contained in the heartbeat, can determine if it is within a trusted network. This state is continuously monitored. In the event that heartbeats are not heard for a prolonged interval of time, the context manager assumes that it is no longer within the trusted network and immediately reverts to the policy prescribed for untrusted networks.

Pietro and Mancini[3] point out that it is important to restrict the web presence of a service to reduce the complexity and traffic for a given network infrastructure. In our design the issued policies are valid only within the scope of the broadcast, that is the hop-count of the broadcast determines how far the heartbeats will be heard. As soon as the user is outside this scope, the policy is no longer valid and the device reverts to the highly restrictive default policy. Thus this mechanism has two effects, viz. it restricts the scope of operation to a particular area, use of granted privileges is disallowed outside this scope and, secondly a context is provided to the device so that only the relevant service interfaces may be exposed in communicating with the handheld device. A device possessing some capabilities allowed by the enforced policy will allow the device to access local services, whereas remote services can always be exposed via proxies if necessary.

Heartbeats are signed by the owner entities and can be verified by other entities involved. Using a PKI infrastructure with X.509 certificates is feasible in this scenario. Trust issues are resolved using CA certificates installed in the mobile device.

Each of the modules shown in figure V.1 are described in the following subsections. Section V.E.1 describes the policy language Rei, section V.E.2 describes the policy engine and section V.E.3 describes the Beacon. Sections V.E.4, V.E.5, and V.E.6 describe the context manager, policy manager, and the policy enforcer respectively.

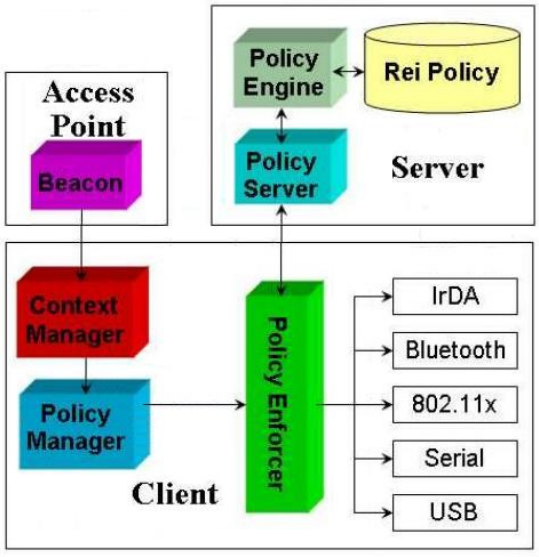

Figure 1. Policy Enforcement Infrastructure

4. Rei Policy Engine and Policy Server

The security policy is described to the Rei Engine using the Rei Ontology. As shown in figure V.2, a domain specific ontology may also be used to describe domain specific information. The Rei policy engine reasons over the policies described to it in the Rei policy language. The Rei Engine has a Java front-end and uses Prolog for its reasoning engine. The role of the Rei engine is to grant access or deny access to requests made by principals in the domain. The policy server is responsible for handling access requests from the various devices in the system, presenting them to the Rei Engine and then distributing these policy certificates to the requesting entities.

The Policy Server first presents the Rei Engine with the current state information of the device in question, which normally includes in the least, the device identifier, the person in possession of the device and the location of the device. The Policy Server then consults the Rei Policy Engine to create a new policy certificate with the granted requests. The policy server then issues this newly created policy certificate to the requesting device. For this particular scenario, the Rei Engine is loaded with the local network acceptable use policy. Later queries to the Rei Engine provide additional information about the location of the device, user of the device etc., when the new policy certificate is to be issued. The policy server issues the request for resource access to the policy engine. The policy engine reasons over the current status of the system and based on the policy for the role of the subject, issues a policy certificate.

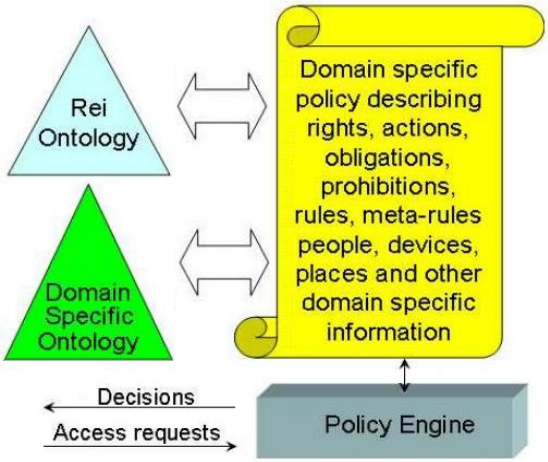

Figure 2. Rei Policy Engine

5. Example Scenarios

5.1. Home Network

Consider the following scenario depicted in figure V.3. Bob is a Ph.D. student affiliated with the lab "A". He has been issued a mobile device that belongs to lab "A". All the lab devices are equipped with the policy enforcement mechanisms described earlier. The lab "A" policy allows all Ph.D students who are affiliated with the lab to be able to use lab "A"'s resources and use the full capabilities of the device they have been leased, while in the "A" lab.

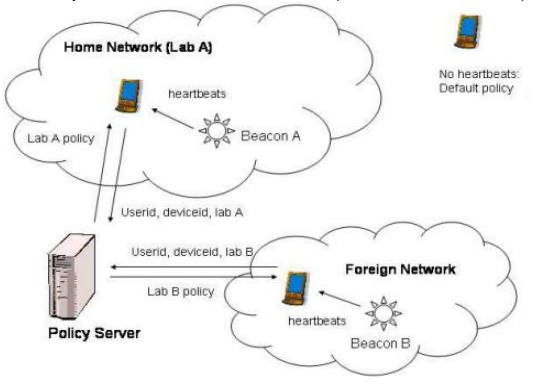

Figure 3. Home network and foreign network

Bob has authenticated himself to the device and, initially the policy enforcer has enforced the default policy on the device, which allows minimal communication. As Bob walks into the "A" lab, the device hears the heartbeats from a beacon. The device verifies that the signature is from one of the beacons it trusts. The context manager then reads the contents of the heartbeat message and signals the policy manager to retrieve the policy server's address, and issue a request for a policy certificate to the policy server. The default policy ensures that such minimal communication is allowed, though other capabilities of the devices remain disabled. The policy server is provided with the device identifier, the user and the location of the device(based on which beacon's heartbeats were heard). The policy server now transforms this information into domain specific information in the Rei language. Then the Rei Engine is queried for access requests based on the device capabilities. A device specific policy certificate is then created, signed by the policy server and issued to the requesting device. The policy manager on the device issues this policy to the policy enforcer. The new policy is then enforced for the time duration specified in the policy certificate. In this case, since Bob is a Ph.D. student and the device was leased to him, he will be able make unrestricted use of the device capabilities within the lab.

When Bob leaves the lab, and is out of range of the beacon, the device can no longer hear the heartbeats. The context manager on the device resets a timer each time it hears a heartbeat. When no heartbeats are heard for a prolonged interval(twice the heartbeat interval), the timer goes off and the context manager resets the device to use the default policy. The policy certificate is valid only for the time interval specified within the certificate and heartbeats from a trusted beacon can be heard.

5.2. Other trusted Networks

Now suppose that Bob, leaves the "A" lab but is still within the university campus and walks into another lab "B" which has a trusted beacon. This lab however has a policy that foreign devices should only be able to use web services via 802.11 but not use IrDA or Bluetooth. This policy may conflict with lab "A"'s policy that all Ph.D. students be allowed unrestricted use of the device's capabilities. However the meta-rules specified in the Rei language can be used to resolve these conflicts. e.g. the meta-rule may resolve the conflict by specifying that the lab policy where the device is present should have priority over all other policies. specifications of this template. You will need to determine whether or not your equation should be typed using either th

6. Summary and Discussion

In this paper we have presented a proof of concept implementation of a policy enforcement infrastructure for mobile devices. We have used a semantic policy language Rei to express security policies. Rei allows policies to be expressed in higher levels of abstraction without requiring knowledge of all possible entities. Policies can be expressed in terms of domain specific information. The policy engine is used to make decisions of allowing or disallowing access requests from actors in the domain.

In our prototype implementation, we demonstrated how a policy can be expressed in the Rei policy language using the Rei Ontology and an augmenting domain specific ontology to describe rights, prohibitions, obligations, dispensations an actor has on the domain actions. We showed how a mobile device equipped with a policy enforcer can be used to dynamically change its behavior and capabilities in a pervasive environment using this security infrastructure. We demonstrated the use of the expressivity of a high level semantic language Rei for describing system wide policies, the dynamic creation of device level policies, policy distribution and, enforcement of these policies on mobile devices.

As noted earlier, the devices with the policy enforcers are themselves trusted devices and cooperate with the security infrastructure. The policy enforcers serve as automatic guards that enforce the correct policy based on current state of the device. This infrastructure addresses security concerns resulting from vulnerabilities in the software or hardware implementations of the device. The security infrastructure does not protect against intentional misuse or attacks.

An alternative to issuing policies from a Policy Server is to use smartcards that contain the policy certificate[4]. The smartcard adds to the hardware requirements of a device. However it is the least obtrusive, since the policy can be enforced so long as the the card monitor notifies the existence of the card. In case of the Policy Server, the sentient program listens for heartbeats from the beacon. It may happen that during periods of severe network congestion heartbeats are lost and the devices suddenly revert to their default policy which will be very disruptive for the users. However in the case of the Policy Server, the policy certificates are created dynamically and are adapted to the context of the device. Also, listening to heartbeats is usually free since most mobile devices come equipped for wireless connectivity, no additional hardware is required. In case of smartcards, the policy is statically issued and stored on the smartcard, it does not change or adapt to changes in a pervasive environment.

An important contribution of the work described in this paper is the actual prototype that has automatic guards based on expressive security policies. In the current implementation there is total reliance on a security infrastructure, e.g., secure beacons whose availability enforce context specific policies, otherwise a restrictive default policy is enforced. Inbuilt tamper-proof hardware security like smartcards can act as secure stores for security policies and can be recharged from time to time. Using an expressive policy language like Rei, these devices can then be configured with generic policies that are adaptive to their current requirements goals based on temporal or spatial contexts.

In subsequent research we will focus our work on how monitoring network conditions, detecting anomalous behavior, measuring availability of trustworthy sources of data streams can be used to adapt to dynamic environmental conditions and resource availability. While the security policies enforced by the security kernels with the help of tamperproof hardware can protect the communication, network interfaces, and data on the device itself, the availability of trusted information sources in a pervasive ecosystem cannot always be assumed.

7. Acknowledgment

This work was financially supported by the Fundamental Research Funds for the Central Universities (No.XDJK2010C044), the Key Program of Chongqing Natural Science(No.CSTC2009BA1006) and the Fundamental Research Funds for the Central Universities (No. XDJK2009C141).

References

[1] Ranganathan, K, "Trustworthy pervasive computing: the hard security problems," In Proceedings of the Second IEEE Annual Conference on Pervasive Computing and CommunicationsWorkshops, 2004, pp.117–121.

[2] J.-P. Hubaux, L. Buttyan, and S. Capkun, "The quest for security in mobile ad hoc networks," In MobiHoc '01: Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing, 2001, pp.146–155.

[3] R. Di Pietro and L. V.Mancini, "Security and privacy issues of handheld and wearable wireless devices," Commun. ACM 46(9), 2003, pp.74–79.

[4] W. A. Jansen, T. Karygiannis, S. Gavrila, and V. Korolev, "Assigning and Enforcing Security Policies on Handheld Devices," In Proceedings of the Canadian Information Technology Security Symposium, May 2002.

[5] J. P. Hubaux, T. Gross, J. Y. L. Boudec, and M. Vetterli, "Towards self-organized mobile ad hoc networks: the Terminodes project," IEEE Communications Magazine 31(1) , 2001, pp.118–124.

[6] Y.-C. Hu, D. B. Johnson, and A. Perrig, "Sead: Secure efficient distance vector routing in mobile wireless ad hoc networks," In Fourth IEEE Workshop on Mobile Computing Systems and Applications (WMCSA '02) , 2002, pp.3–13.

[7] Y.-C. Hu, A. Perrig, and D. B. Johnson, "Ariadne: a secure on-demand routing protocol for ad hoc networks," In Proceedings of the 8th annual international conference on Mobile computing and net-working, 2002, pp.12–23.

[8] Y.-C. Hu, A. Perrig, and D. B. Johnson, "Rushing attacks and defense in wireless ad hoc network routing protocols," In Proceedings of the 2003 ACM workshop on Wireless security, 2003, pp.30–40.

[9] A. Dix, T. Rodden, N. Davies, J. Trevor, A. Friday, and K. Palfreyman, "Exploiting space and location as a design framework for interactive mobile systems," ACM Trans. Comput.-Hum. Interact., 7(3) , 2000, pp.285–321.

[10] M. J. Covington,W. Long, S. Srinivasan, A. K. Dey,M. Ahamad, and G. D. Abowd, "Securing context-aware applications using environment roles," In SACMAT '01: Proceedings of the sixth ACM symposium on Access control models and technologies, 2001, pp.10–20.

[11] L. Buttyan and J.-P. Hubaux, "Enforcing service availability in mobile ad-hoc wans," In MobiHoc '00:Proceedings of the 1st ACM international symposium on Mobile ad hoc networking & computing, IEEE Press, 2000, pp.87–96.

[12] A. Aziz andW. Diffie, "Privacy and Authentication forWireless Local Area Networks," IEEE Personnal Communications, 1(1), 1993, pp.25–31.

[13] J. B. Begole, J. C. Tang, R. B. Smith, and N. Yankelovich. Work rhythms: analyzing visualizations of awareness histories of distributed groups. In CSCW '02: Proceedings of the 2002 ACM conference on Computer supported cooperative work, 2002, pp.334–343.

[14] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The eigentrust algorithm for reputation management in p2p networks," In WWW '03: Proceedings of the 12th international conference on World Wide Web, 2003, pp.640–651.